

YSpotify API

Service d'extensions de l'API Spotify

Prérequis

1. Posséder un compte Spotify ;
2. Avoir créé un client Spotify sur votre [espace développeur \(Dashboard\)](#).

Lexique

Service : Application exposant les fonctionnalités.

Utilisateur : Utilisateur **inscrit** et **authentifié** sur le Service.

Utilisateur anonyme : Utilisateur **non-authentifié** sur le Service.

Groupe : Ensemble d'Utilisateurs.

Chef : Utilisateur ayant des droits privilégiés sur son Groupe.

Fonctionnalités obligatoires

Cette section décrit l'ensemble des fonctionnalités **obligatoires** que le Service doit exposer.

FT-1 Inscription

L'Utilisateur anonyme peut s'inscrire sur le Service. Il doit alors fournir les informations suivantes :

- Un nom d'utilisateur (pseudo) ;
- Un mot de passe.

Le nom d'utilisateur de chaque Utilisateur sert d'**identifiant** et doit donc être **unique** au sein du Service. Autrement dit, deux Utilisateurs ne peuvent pas avoir le même pseudo.

Quant au mot de passe, aucune contrainte n'est imposée.

FT-2 Connexion

L'Utilisateur anonyme peut se connecter au Service en fournissant son pseudo et mot de passe. Il obtient alors un token d'accès au Service, l'authentifiant pour toutes ses futures requêtes. Le token a une durée de validité d'**une heure**.

FT-3 Liaison du compte Spotify

L'Utilisateur peut déléguer l'autorisation au Service d'utiliser les services (API) de Spotify en son nom.

Plus de détails sont donnés plus bas dans la section [Authentification](#).

FT-4 Rejoindre un Groupe

L'Utilisateur peut rejoindre un Groupe. Les Groupes sont identifiés simplement par leur nom (chaîne de caractères).

Si l'Utilisateur tente de rejoindre un Groupe qui n'existe pas, alors le Groupe est automatiquement créé et l'Utilisateur en devient le Chef.

Si l'Utilisateur appartient déjà à un Groupe, alors il le quitte **automatiquement** avant de rejoindre le nouveau. Si l'Utilisateur quittant le Groupe est le Chef, un nouveau Chef est aléatoirement assigné parmi les membres restants. S'il ne reste plus personne dans le Groupe, le Groupe n'existe plus.

N.B. : Par défaut (à l'inscription), un Utilisateur n'appartient à aucun Groupe.

FT-5 Consultation des Groupes et Utilisateurs

Liste des Groupes

L'Utilisateur peut consulter la liste de tous les Groupes existant sur le Service. Les informations retournées associées à chaque Groupe sont les suivantes :

- Nom du groupe ;
- Nombre d'Utilisateurs dans le groupe.

Liste des membres du Groupe

L'Utilisateurs peut également consulter la liste de tous les Utilisateurs appartenant à **son** Groupe. Les informations retournées associées à chaque Utilisateur sont les suivantes :

- Nom d'utilisateur (pseudo) ;
- S'il est le Chef du Groupe ;
- Pseudo du compte Spotify associé (si liaison) ;
- Morceau en cours d'écoute (si liaison) ;
 - *Titre du morceau, nom de l'artiste, titre de l'album ;*
- Nom de l'appareil d'écoute actif (si liaison).

Un Utilisateur n'appartenant à aucun Groupe n'est pas autorisé à effectuer cette requête.

FT-6 Personnalité de l'Utilisateur

Le Service doit être capable de déduire la personnalité de l'Utilisateur effectuant la requête en fonction de ses "Titres Likés". Plus précisément, tous les "Titres Likés" sont analysés¹ pour générer un portrait de l'Utilisateur, contenant les informations suivantes (liste exhaustive) :

- Attrait pour la dance (entier de 0 à 10) ;
- Agitation (tempo moyen écouté) ;
- Préférence entre les musiques vocales ou instrumentales ;
- Attitude plutôt positive ou négative (*voir l'attribut "valence"*).

Si l'Utilisateur ne dispose d'aucun titre dans ces "Titres Likés", la ressource (personnalité de l'utilisateur) n'existe pas.

FT-7 Synchronisation

Le **Chef** peut synchroniser la musique qu'il est en train d'écouter sur tous les appareils actifs des autres Utilisateurs (Utilisateurs synchronisés) **appartenant à son Groupe**. Le périphérique actif de chaque Utilisateur synchronisé joue alors la musique exactement à la même position (temps) que l'Utilisateur synchronisant au moment où celui-ci a effectué la requête de synchronisation.

FT-8 Playlist

L'Utilisateur peut demander la création d'une playlist sur son compte Spotify contenant les 10 musiques préférées² d'un autre Utilisateur (qui peut être lui-même) passé en paramètre.

¹ <https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features>

² <https://developer.spotify.com/documentation/web-api/reference/#/operations/get-users-top-artists-and-tracks>

L'Utilisateur passé en paramètre doit appartenir au même Groupe que l'Utilisateur ayant effectué la requête.

Spécifications optionnelles

Cette section décrit l'ensemble des spécifications (fonctionnelles et techniques) **optionnelles**.

FT-9 Tests et qualité

Chaque fonctionnalités (FT-1 à FT-8) pouvant être testées doivent l'être par une batterie de tests unitaires et d'intégration.

FT-10 Conteneurisation

Le Service doit pouvoir être déployé dans un conteneur Docker.

FT-11 Reverse proxy

Une configuration doit être fournie permettant de faire tourner le Service derrière un reverse proxy.

Authentification

Les fonctionnalités offertes par notre service (*application*) utilisent des *ressources* possédées par l'Utilisateur (*resource owner*). Afin que notre service puisse accéder à ces ressources protégées, l'Utilisateur doit s'authentifier auprès de Spotify et **déléguer l'autorisation** au service pour effectivement l'autoriser à y accéder.

Heureusement, l'API de Spotify supporte OAuth 2.0 pour la délégation des autorisations³.

N.B. : L'Utilisateur doit forcément, à un moment ou un autre, accéder à une URL dans son navigateur et effectuer une action manuelle afin d'autoriser notre service/client à accéder à ses ressources.

Autorisation

Les FT-1 et FT-2 sont les seules fonctionnalités accessibles par un Utilisateur anonyme.

Les FT-6, FT-7, FT-8 nécessitent que l'Utilisateur ait lié son compte Spotify, sans quoi l'accès lui est refusé.

Documentation

L'ensemble de votre API doit être documenté grâce à la spécification OpenAPI (dernière release). Un SwaggerUI doit être accessible à `http://localhost:<port>/api-docs`.

Toutes les fonctionnalités de l'API doivent pouvoir être testées directement via le SwaggerUI généré.

Un fichier **README** doit être présent à la racine de votre projet, décrivant toutes les procédures importantes pour faire fonctionner votre API (installation, lancement, ...) ainsi que la **composition de votre équipe**.

Contraintes et détails techniques

Aucun framework, bibliothèque n'est imposé.

³ .<https://developer.spotify.com/documentation/general/guides/authorization/>
<https://developer.spotify.com/documentation/general/guides/authorization/implicit-grant/>

Authentification

Le Service n'implémente pas OAuth 2.0 ! Aucun mécanisme propre à OAuth 2.0 (Refresh Token, ...) ne doit être mis en place.

En revanche, vous avez le choix d'utiliser ou non les JWT pour les tokens émis.

Persistance des données

La totalité des données nécessaires doit être stockée dans un seul fichier nommé "users.json".

Les mots de passes doivent être stockés hachés (SHA256).

Modalités d'évaluation

Zone	Détails	Coefficient
Fonctionnel	Respect des spécifications fonctionnelles	3
Technique	Respect des bonnes pratiques et des spécifications liées aux protocoles, standards et technologies utilisées pour ce projet	2
Projet	Bon déroulement du projet. Gestion des priorités, ressources, temps, ...	2
Documentation	Documentation claire, complète et fonctionnelle (SwaggerUI)	1

La note finale est commune à l'ensemble de l'équipe.

Les spécifications optionnelles, si bien implémentées, sont un **bonus** à la note. Autrement dit, il est possible d'avoir 20 même sans introduire ces fonctionnalités.

La note finale est plafonnée à 20/20.

Rendus / Livrables

Code source

Le code source doit contenir tous les éléments nécessaires au lancement de votre projet, incluant la documentation (SwaggerUI !) et éventuellement les tests.

"Projet"

Synthèse du déroulement du projet du point de vue de votre équipe. Seulement le côté "Gestion" est attendu. Plus précisément, les éléments suivants doivent être expliqués en détails :

- Comment avez-vous décomposé le projet ? Quelles ont été les phases (documentation, développement, tests, ...) ?
- Comment avez-vous reparti la charge en fonction des ressources (membres de l'équipe, prof pour répondre aux questions, ...) ? Qui s'est occupé de quoi ?
- Comment avez-vous géré votre temps ? Avez-vous défini des priorités ?
- Avez-vous défini une stratégie pour vous documenter ?

- Avez-vous produit des documents pour mieux comprendre le sujet (schémas, user stories, ...) ?

D'une manière générale, **tous vos choix doivent être justifiés !**

Ce rendu peut être fait soit sous forme de rapport (document Word, pdf, ...), soit sous forme d'oral. Ces deux formes de rendu sont décrites ci-dessous.

Rapport de Projet (option 1)

Un rapport d'environ 5 pages est attendu. Format PDF uniquement.

La nature de ce format vous permet de décrire quelques choix techniques (**importants seulement**) que vous avez dû faire.

Soutenance de Projet (option 2)

Vous devez préparer un oral où tous les membres de l'équipe passeront devant la classe. L'oral doit durer environ 5 min. Un support (slides) n'est pas obligatoire.

La Soutenance **n'est pas** une démonstration de votre Service mais une justification du travail qui a été fait (selon les éléments décrits précédemment).

Ressources utiles...

Description des endpoints	https://developer.spotify.com/documentation/web-api/reference/#/
Liste complète des scopes exposés par l'API Spotify	https://developer.spotify.com/documentation/general/guides/authorization/scopes/
Guide sur l'Authorization Code Flow de Spotify	https://developer.spotify.com/documentation/general/guides/authorization/code-flow/